Fundamentals of Computer Programming

Building a Programming Portfolio

Week 8

You should be able to complete the following programs by the end of the week. By now you should understand why you should be saving your work to GitHub or similar. Possible solutions will be uploaded to the main module GitHub repository every week. If you follow that repo you should be able to receive notifications.

We are near the end of the module.So let's write some programs that could genuinely be useful.

1. The Unix `nl` command prints the lines of a text file with a line number at the start of each line. (It can be useful when printing out programs for dry runs or white-box testing). Write an implementation of this command. It should take the name of the files as a command-line argument.
2. The Unix `diff` command compares two files and reports the differences, if any. Write a simple implementation of this that takes two file names as command-line arguments and reports whether or not the two files are the same. (Define "same" as having the same contents.)
3. The Unix `grep` command searches a file and outputs the lines in the file that contain a certain pattern. Write an implementation of this. It will take two command-line arguments: the first is the string to look for, and the second is the file name. The output should be the lines in the file that contain the string.
4. The Unix `wc` command counts the number of lines, words, and characters in a file. Write an implementation of this that takes a file name as a command-line argument, and then prints the number of lines and characters.
   Note: Linux (and Mac) users can use the "`wc`" command to check the results of their implementation.
5. The Unix `spell` command is a simple spell-checker. It prints out all the words in a text file that are not found in a dictionary. Write and test an implementation of this, that takes a file name as a command-line argument.
   Note: You may want to simplify the program at first by testing with a text file that does not contain any punctuation. A complete version should obviously be able to handle normal files, with punctuation.
   Another Note: You will need a list of valid words. Linux users will already have one (probably in `/usr/share/dict/words`). It is more complicated, as usual, for Windows users. Happily, there are several available on GitHub.